

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Colarič

Digitalizacija in avtomatizacija vrta

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Robert Rozman

Ljubljana 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Danes se avtomatizacija in informatizacija procesov uporabljata praktično na vseh področjih. Zdrava prehrana in okolju prijazna pridelava poljskih in vrtnih pridelkov postajata vedno bolj pomembni komponenti trajnostnega razvoja. V nalogi raziščite možnosti digitalizacije in avtomatizacije vrtnih površin. Na tej osnovi realizirajte celostni prototipni sistem, ki bo omogočal optimalno upravljanje vrtnih površin. Sistem naj s pomočjo mobilnih in avtonomnih modulov spremlja dejavnike v okolju in glede na zbrane podatke omogoči optimalne pogoje za rast in razvoj rastlin z aktivno regulacijo zalivanja. Senzorski moduli naj omogočajo brezžično komunikacijo z osrednjim delom sistema, ki zbira podatke in jih posreduje v spletni strežnik. Ob tem poskusite zagotoviti čim daljšo časovno avtonomijo senzorskih modulov in njihov čim daljši komunikacijski doomet. Sistem naj aktivno upravlja zalivanje površin tako, da za zagotavljanje optimalnih pogojev porabi čim manj vode. Tako doomet senzorskih modulov kot optimalnost načina zalivanja preizkusite tudi v praksi. Ob vsem tem poskusite doseči čim boljše cenovno dostopnost in enostavnost uporabe sistema, tudi na daljavo preko spleta.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Luka Colarič sem avtor diplomskega dela z naslovom:

Digitalizacija in avtomatizacija vrta

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Roberta Rozmana,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 25. februarja 2016

Podpis avtorja:

Zahvaljujem se mentorju za pomoč, staršem za podporo in ostalim za potrpljenje.

Mami za navdih nad vrtnarstvom.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Sodobno upravljanje agrikulturnih površin	3
2.1	Pomembni parametri za nadzor agrikulturnih površin	4
2.2	Agrikulturni testni prostor	6
3	Gradniki sistema	9
3.1	Izbrani senzorji in moduli	9
3.2	Arduino razvojne ploščice	13
3.3	Raspberry Pi računalniški sistem	14
4	Delovanje sistema	17
4.1	Osnovni gradniki sistema	17
4.2	Oddaljeni moduli	19
4.3	Glavni modul	25
5	Postavitev strežnika in spletne strani	31
5.1	Strežnik	31
5.2	Spletna stran	35

KAZALO

6	Preizkus sistema v praksi	41
6.1	Primerjalni preizkus različnih načinov zalivanja	41
6.2	Testiranje dometa oddaljenih modulov	51
6.3	Poraba oddaljenih modulov	51
7	Sklepne ugotovitve	55
	Literatura	57

Slike

2.1	Fujitsu oblak za nadzor agrikulture	4
2.2	Testni prostor - vikend na Gorjancih	7
3.1	Senzor za merjenje vlage in temperature DHT-11	10
3.2	Senzor za merjenje vlažnosti zemlje FC-28	11
3.3	Foto upor LDR05	11
3.4	Senzor zračnega pritiska BMP085	12
3.5	Brezžični modul nRF24L01	12
3.6	Prikaz sistemov Arduino uno na levi in Arduino nano na desni	14
3.7	Raspberry Pi	15
4.1	Splošen prikaz delovanja sistema	18
4.2	Shema vezja oddaljenega modula	20
4.3	Izdelan oddaljeni modul s senzorji	21
4.4	Posplošen prikaz algoritma za zalivanje	23
4.5	Vezalna shema glavnega modula	28
4.6	Izdelan glavni modul	29
5.1	Primer elektronskega sporočila	34
5.2	Spletna stran s podatki glavnega modula in trenutne vremen- ske razmere	36
5.3	Spletna stran z grafom poteka okoljskih parametrov	37
5.4	Zemljevid lege in trenutne vrednosti okoljskih parametrov vrta	38
5.5	Najvišje in najnižje vrednosti podatkov	38

5.6	Vnos in sprememba podatkov	39
6.1	Primerjava vseh testnih lončkov	42
6.2	Sobni testni prostor zalivalnega sistema našega projekta	43
6.3	Končna izvedba testnega namakalnega sistema s peristaltično črpalko	44
6.4	Testni lonček A	45
6.5	Testni lonček B	47
6.6	Testni lonček C	48
6.7	Testni lonček D	50
6.8	Vizualizacija dometov obeh brezzičnih modulov	52

Seznam uporabljenih kratic

kratica	angleško	slovensko
RAM	Random Access Memory	bralno-pisalni pomnilnik
ROM	Read Only Memory	bralni pomnilnik
V	Volt	volt
A	Ampere	amper
LDR	Light Dependent Resistors	svetlobno občutljivi upor
USB	Universal Serial Bus	univerzalno serijsko vodilo
PCB	Printed Circuit Board	tiskano vezje
IDE	Integrated Development Environment	vgrajeno razvojno okolje
Hz	Hertz	herc
REST	Representational State Transfer	spletni protokol
I2C	Inter-Integrated Circuit	komunikacija med integriranimi vezji

Povzetek

Namen diplomskega dela je razvoj sistema za digitalizacijo in avtomatizacijo agrikolturnih oziroma vrtnih površin. Sistem je sestavljen iz treh glavnih delov. Oddaljeni moduli imajo priključene senzorje za temperaturo in vlago zraka, vlažnost zemlje ter osvetljenost okolice. Oddaljeni moduli imajo tudi funkcijo avtomatičnega zalivanja. Glavni modul sprejema podatke iz oddaljenih modulov in jih posreduje strežniku. Na strežniku je podatkovna baza za shranjevanje podatkov in spletna stran za pregled teh podatkov.

V diplomski nalogi so najprej navedeni pomen in razvoj digitalizacije agrikulture ter dejavniki, ki so potrebni za rast in razvoj rastlin. Sledi opis sistema digitalizacije in avtomatizacije - izbira senzorjev in razvojnih ploščic, delovanje glavnih in oddaljenih modulov ter algoritem zalivanja. Opisani so strežnik, spletna stran in podatkovne baze. Naloga se zaključi s praktično demonstracijo delovanja izdelanega sistema.

Ključne besede: digitalizacija, avtomatizacija, Arduino nano, Raspberry Pi, senzorji, strežnik, podatkovna baza, spletna stran, agrikultura.

Abstract

The purpose of this thesis is a development of system for digitization and automation of garden surfaces. System consists of three main parts. Remote modules have connected sensors for temperature, air humidity, soil humidity and environment luminosity. Remote modules also have automatic watering feature. Main module receives data from remote modules and sends them to the server. On server there is database for saving the data and web page for visualization.

In thesis, the meaning and development of agriculture digitization are first described including factors that are important for plants growth and healthy development. In next section the implementation details of the system for digitization and automation are presented - particularly selection of sensors and development boards, workings of main and remote modules and watering algorithm. The central part of the system consists of server, web page and database. Thesis ends with practical demonstration and analysis of the working system.

Keywords: digitization, automation, Arduino nano, Raspberry Pi, sensors, server, database, web page, agriculture.

Poglavje 1

Uvod

V dobi hitro spreminjajočega sveta se spreminjajo tudi najbolj osnovne človekove dejavnosti, kot je pridelava sadja in zelenjave. S tem povezani postopki se poskušajo avtomatizirati, okoljski parametri pa zajeti z računalniki oziroma digitalizirati. Podatki digitalizacije se selijo na splet. Tam se spremljajo in iz njih razbirajo razmere zraka, sonca in zemlje, kjer se vzgajata sadje in zelenjava. Iz teh podatkov je razvidno, kako se razmere spreminjajo preko dneva, skozi letne čase in kako se skozi leta spreminjajo podnebne razmere.

Na začetku je bil namen avtomatizacije poljedelskih in vrtnih površin zmanjšati in poenostaviti delo pridelovalcev in oskrbnikov. Sistemi so bili preprosti in nepovezani, namenjeni so bili predvsem intervalnemu zalivanju površin, kar pa ni bilo vedno usklajeno s pomembnimi zunanjimi dejavniki, kot sta trenutna namočenost zemlje in osončenost zemljišča. Vpliv enega in drugega je lahko za rastline uničujoč, v prvem primeru povzroči gnitje, v drugem učinek povečevalnega stekla na moč sončnih žarkov.

Kasneje so se sistemi izboljšali in začeli upoštevati zunanje dejavnike ter se jim prilagajati. Nadzorovali so izvajanje različnih potrebnih aktivnosti - poleg zalivanja so ogrevali tople grede, optimizirali vlago v zraku, zastirali sonce s premičnimi strehami, ...

Agrikultura je največja in za človekov obstoj ena izmed najbolj pomembnih panog. Pod besedo agrikulturo spadata kmetijstvo in poljedelstvo. Z

namenom kvalitetne avtomatizacije in digitalizacije smo izdelali sistem, ki je prilagodljiv glede na velikost in število agrikolturnih površin, prav tako pa sistem uporabniku na prijazen način ponuja podatke, ki mu omogočajo kvalitetno vzgojo zdravega sadja in zelenjave.

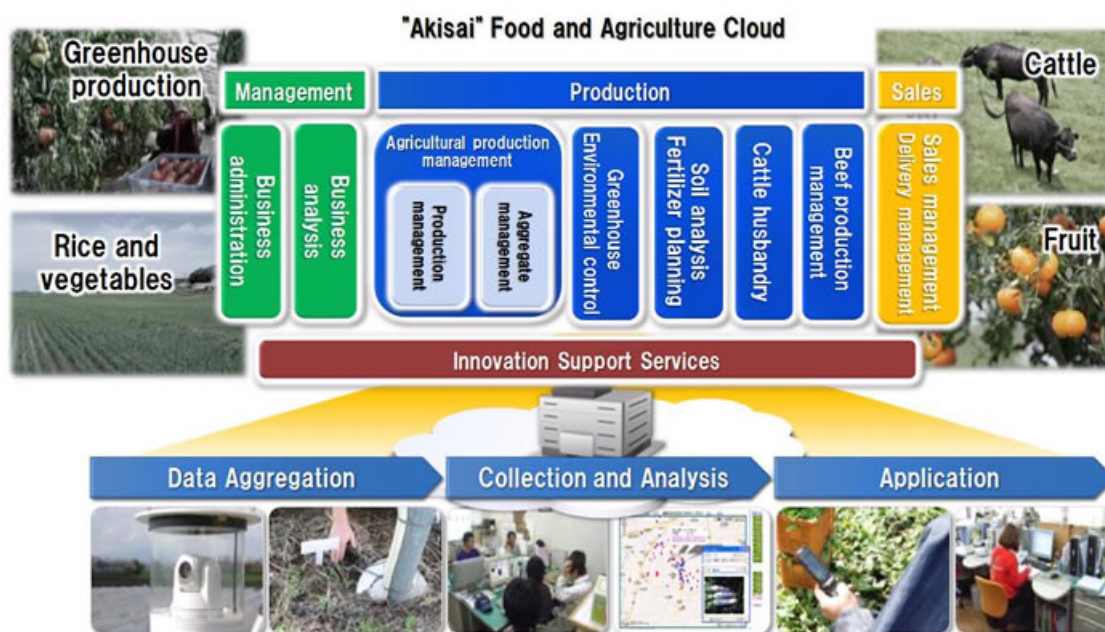
V diplomski nalogi je na začetku opisan razvoj in pomen digitalizacije agrikulture. V nadaljevanju so predstavljeni parametri izdelanega sistema za avtomatizacijo in digitalizacijo agrikolturne dejavnosti ter kako so uporabniku prikazani. Kasneje je podrobno razloženo delovanje sistema. V tretjem delu je opisana podatkovna baza, kjer se podatki senzorskih modulov shranjujejo. Pojasnjeno je tudi kako se dostopa do podatkovne baze in spletna stran, ki je vidna uporabniku. Ta mu prikazuje podatke iz njegovega vrta. Na koncu pa je opisan preizkus delovanje sistema v realnem okolju, kateremu je sistem namenjen. Povzete so tudi možne izboljšave in ugotovitve, ki so nastale tekom celotnega projekta.

Poglavje 2

Sodobno upravljanje agrikulturnih površin

Kot že v uvodu omenjeno, se veliko osnovnih dejavnosti seli v digitalno dobo in agrikultura ni izjema. Razvoj pametnih načinov za nadzor kmetijskih površin se je začel precej kasneje kot je to omogočala tehnologija. Začetki digitalizacije segajo v industrijo, malo kasneje se pojavijo "pametni domovi". Šele leta 2012 je Fujitsu [1] postavil platformo za inteligentno vodenje agrikulturnih površin, ki je bila na začetku namenjena nadzoru riževih polj in je omogočala shranjevanje temperature in vlage v oblak. Beseda oblak predstavlja spletno shrambo podatkov, ki jih lahko shranjujemo, obdelujemo in bremo z napravami, povezanimi v splet. Hkrati je omogočala oddaljen dostop prižiganja grelcev zraka za nadzor temperature in prezračevalnikov za nadzor vlage. Ti sistemi so zaživi zaradi preprostih razlogov, kot je zmanjšanje števila zaposlenih, prav tako pa je implementacija te platforme omogočala preprost pregled podatkov, saj so bili vzeti iz senzorjev, postavljenih na agrikulturnih površinah in prikazani na ekranu. Te podatke so tudi preprosto uporabljali v analizah in izračunavanju trendov. Podatki so ažurni in brez napak. Slika 2.1 [1] prikazuje splošen pregled te platforme. Na sliki so prikazane storitve njihovega oblaka. Ta uporabnikom prikazuje na strnjen način vse vidike vodenja agrikulture. Med te vidike spada administracija, proizvo-

dnja in prodaja. Kot prikazano na sliki, se podatki prvotno zbirajo preko senzorjev. Nato se analizirajo in na koncu uporabijo za izboljšanje agrikulturnih dejavnosti.



Slika 2.1: Fujitsu oblak za nadzor agrikulture

2.1 Pomembni parametri za nadzor agrikulturnih površin

Pri izbiri parametrov, ki smo jih želeli nadzorovati na agrikulturnih površinah, smo se odločali na podlagi dveh kriterijev pomembnosti: za rast in razvoj rastlin in obstoju senzorjev za merjenje tega parametra. Npr. parameter pH (kislost ali bazičnost zemlje) je pomemben za zdrav razvoj rastline, a senzor, ki bi ga meril, ni preprost za uporabo. Potrebno ga je namreč vsake toliko časa kalibrirati z različnimi raztopinami [2], nezanemarljiva pa je tudi izrazito višja cena v primerjavi z ostalimi senzorji. Izbrani parametri, ki so bili smiselni pri našem projektu, so opredeljeni v nadaljevanju.

2.1.1 Svetloba

Svetloba spada med najpomembnejše dejavnike za rast in zdrav razvoj vsake rastline. Rastlina s pomočjo svetlobe ustvarja fotosintezo, biokemijski proces, ki ji omogoča, da le-to pretvori v energijo za lastno rast [3]. Pri svetlobi so za rastlino pomembni trije faktorji [4]:

1. **Kvaliteta svetlobe** za rastlino pomeni barvo svetlobe. Rdeč in moder spekter sta za rastlino najbolj pomembna in ju tudi največ sprejme. Zelen spekter pa je zanjo najmanj uporaben in ga v večini odbije.
2. **Količina svetlobe** je v naravi odvisna od letnega časa, od geografske lege in vremenskih razmer. Večja kot je količina svetlobe, bolj intenzivno v rastlini poteka proces fotosinteze.
3. **Trajanje svetlobe** predstavlja časovno izpostavljenost rastline svetlobi. Ta dejavnik v rastlini sproža različne cikle. Če je rastlina dlje časa izpostavljena svetlobi, bo rastla, krajša izpostavljenost pa v rastlini sproža cvetenje in posledično nastajanje plodov.

2.1.2 Temperatura

Temperatura je prav tako eden izmed nepogrešljivih dejavnikov za rastlino. Večina rastlin uspeva med 0°C in 50°C. Optimalna temperatura skozi dan, skozi noč in najvišja še sprejemljiva temperatura za rastlino pa so različne za posamezne rastline. Aktivnost encimov in kemičnih reakcij v rastlini se večja z višanjem temperature, približno za faktor dva za vsakih 10°C, vendar le do določene temperature. Previsoke temperature povzročijo razpad encimov in proteinov v rastlini.

Pretirano nizke temperature prav tako škodujejo rastlini in omejujejo njeno rast. Z nižanjem temperature se manjša absorpcija vode v rastlino zaradi višje viskoznosti vode pri nižjih temperaturah. Temperature pod 0°C pa povzročijo ekspanzijo v rastlini, kar uniči celične membrane.

2.1.3 Vlaga zraka

Vlaga zraka je količina razpršene vode v zraku, ta količina pa je povezana s temperaturo zraka. Toplejši zrak ima zmožnost zadržati več vode kot hladnejši. Ta vrednost se prav tako spreminja za faktor dva. Npr. pri 20°C je dvakrat manj vlage v zraku, kot pri 10°C.

Vlaga zraka vpliva na odprtost listnih rež rastline, ki uravnavajo izgubo vode iz rastline preko kemičnih reakcij, podobnim znojenju človeka.

2.1.4 Vlaga zemlje

Vlaga zemlje je količina vode v zemlji, prav tako zelo pomembna za obstoj rastline. V primeru, da je vlaga zemlje visoka dlje časa, obstaja nevarnost, da rastlina zgine. Če je vlage premalo, pa rastlina ne more vsrkati dovolj vode in odmre zaradi pomanjkanja. Kvalitetna zemlja omogoča, da se voda normalno porazdeli po njej, višek pa odteče; tako odigra bistveno vlogo pri optimalni namočenosti tal.

V projektu smo zato še posebno pazili na ta dejavnik. Pravilna namočenost zemlje rastlini omogoča, da lahko vsrka dovolj hranilnih substanc. Pri tem pa pomembno vlogo igra kvaliteta prsti, v kateri bo rastlina rasla.

Načeloma poznamo tri vrste prsti: peščeno, ilovnato in glineno. Ilovnata prst je najbolj zaželjena, saj jo sestavljajo približno enaki delci peska, glin in organskih snovi. Ilovnata tla pozimi redko zadržujejo preveč vode, poleti pa se redko izsušijo. So najprimernejša za agrikulturne površine oziroma za rastline [5].

2.2 Agrikulturni testni prostor

Za testni prostor smo izbrali naš vikend na Gorjancih (Slika 2.2), ki ima nekaj zelenjavnih in okrasnih vrtov ter sadovnjak. Primeren je bil zaradi možnosti primerjave med vrtovi, v hiški pa je bila nameščena glavna postaja, povezana na električno napeljavo in splet.



Slika 2.2: Testni prostor - vikend na Gorjancih

Ker je projekt nastajal v hladnejšem delu leta, na testnem prostoru nismo mogli testirati sistema za zalivanje. Bil pa je zelo uporaben za testiranje dometov naših oddaljenih modulov. Del projekta za avtomatsko zalivanje smo zato izvajali v manjšem testnem protoru v sobi. V lončnice smo posadili čili in smo ga nato imeli ves čas pod nadzorom.

Poglavje 3

Gradniki sistema

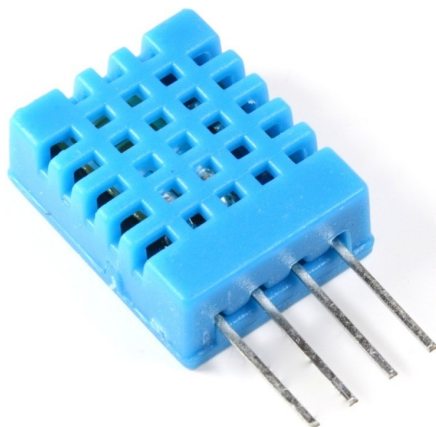
V tem poglavju bomo opisali gradnike, ki smo jih izbrali v našem sistemu. Vsi gradniki so bili izbrani na podlagi različnih kriterijev oziroma razlogov, ki so prav tako pojasnjeni v tem poglavju.

3.1 Izbrani senzorji in moduli

Kot omenjeno v poglavju 2, smo senzorje izbirali glede na pomembnost parametra za rast in razvoj rastline. Pri dejanski izbiri senzorjev pa smo se odločali glede na čim večje območje napajanja in cenovno ugodnost ter s tem tudi zmanjšali končno ceno projekta. Na podlagi tega smo izbrali naslednje senzorje in module:

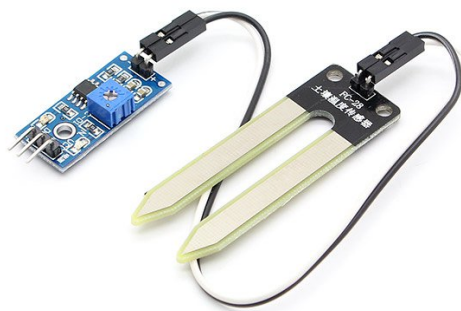
- **Senzor temperature in vlage DHT-11** (Slika 3.1) je nizkocenovni senzor za merjenje temperature in vlage zraka v enem. Za merjenje vlage uporablja kapacitivni senzor vlage, za merjenje temperature termistor. Senzor ima območje napajanja med 3V in 5V, tako da je priročen za različne mikrokontrolerje. Najvišja poraba senzorja je 2.5mA. Območje merjenja za vlago je med 20-80% z natančnostjo 5%. Območje merjenja temperature je od 0°C do 50°C z natančnostjo $\pm 2^\circ\text{C}$. Največja hitrost zajemanja podatkov je 1Hz, kar je dovolj za naše potrebe. Ta senzor predstavlja t.i. ozko grlo, saj je najpočasnejši od

vseh uporabljenih senzorjev. Zato čas zajema ne more biti pogostejši kot enkrat na sekundo. [6]



Slika 3.1: Senzor za merjenje vlage in temperature DHT-11

- **Senzor vlažnosti zemlje FC-28** (Slika 3.2) je senzor za merjenje vlažnosti zemlje, prav tako nizkocenov. Vlažnost zemlje meri glede na upornost med nožicama. Kadar je vlažnost zemlje večja, se zmanjša upor v senzorju. Območje napajanja je med 3V in 5V [7]. Natančnost merjenja se veča z večanjem števila vzorcev oziroma podatkov iz senzorja in računanja povprečja. Upornost se spreminja precej hitro na začetku zalivanja rastline, ko se voda v zemlji še ni porazdelila in ustalila. Pri dolgotrajni uporabi pa so nastale manjše težave, ker so nožice senzorja začele rjaveti. Ta problem pa ni vplival na natančnost merjenja.
- **Foto upor LDR05** (Slika 3.3) je upor, ki spreminja upornost glede na količino svetlobe, ki pade nanj. V temi ima foto upor zelo veliko upornost in sicer do 1M ohma. Ko pa je izpostavljen svetlobi, mu upornost zelo pade, na nekaj ohmov. Ko na upor posije svetloba, se upornost ne spremeni v trenutku. Ta pojav se imenuje hitrost okrevanja upornosti (angleško resistance recovery rate). Zakasnitev traja



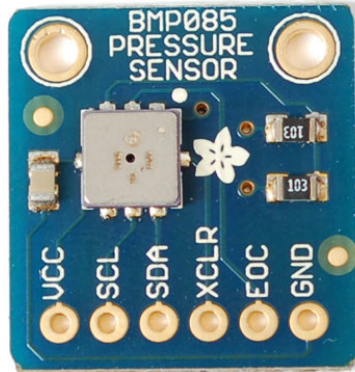
Slika 3.2: Senzor za merjenje vlažnosti zemlje FC-28

10ms, kar je priročno, saj se lahko upornost v foto uporu spreminja zelo hitro. Izračunamo povprečje podatkov, izmerjene v eni sekundi in tako dobimo bolj konstantne podatke [8].



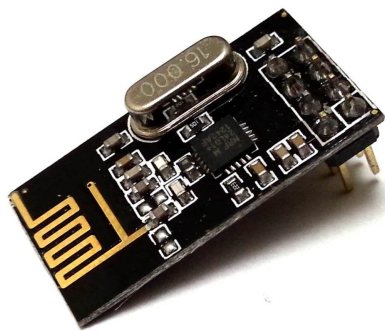
Slika 3.3: Foto upor LDR05

- **Senzor zračnega pritiska BMP085** (Slika 3.4) je senzor, ki meri zračni pritisk. Za povezavo uporablja I2C vodilo, območje napajanja pa je med 1.8V in 3.6V. Razpon merjenja, ki ga omogoča senzor je med 300hPa in 1100hPa z natančnostjo 0.02hPA. Ker se pritisk manjša z višino, lahko pretvorimo, da je senzor uporaben med +9000m pa vse do -500m nadmorske višine. Poraba senzorja je približno 5uA pri frekvenci zajemanja 1Hz [9].



Slika 3.4: Senzor zračnega pritiska BMP085

- **Modul nRF24L01** (Slika 3.5) je brezžični modul, ki smo ga v našem sistemu uporabili za komunikacijo med oddaljenimi moduli, postavljenimi na vrtove in glavnim modulom, ki je te podatke sprejemal in jih pošiljal na strežnik. Modul deluje na frekvenci 2.4GHz in ima hitrost oddajanja podatkov do 1Mbps, kar je bilo več kot preveč za naš projekt. Modul deluje na napetosti od 1.9V do 3.6V. Najvišja poraba, ki jo doseže pri oddajanju in sprejemanju podatkov je približno 14mA [10].



Slika 3.5: Brezžični modul nRF24L01

3.2 Arduino razvojne ploščice

Pri našem projektu smo za mikrokrmilnik izbrali Arduino platformo. Ta v našem sistemu opravlja funkcije oddaljenih modulov. Razlogov za to je bilo več. Ker je Arduino odprto koden, obstaja več brezplačnih razvojnih orodij za programiranje Arduina. Prav tako obstaja veliko vrst Arduino razvojnih ploščic, ki so primerne za vsak projekt. Ploščice se razlikujejo po velikosti, številu priključkov, ki so dostopni in različni funkcionalnosti, ki jih nudijo nekatere ploščice, npr. priključek za ethernet kabel, ki omogoča dostopanje Arduina na splet.

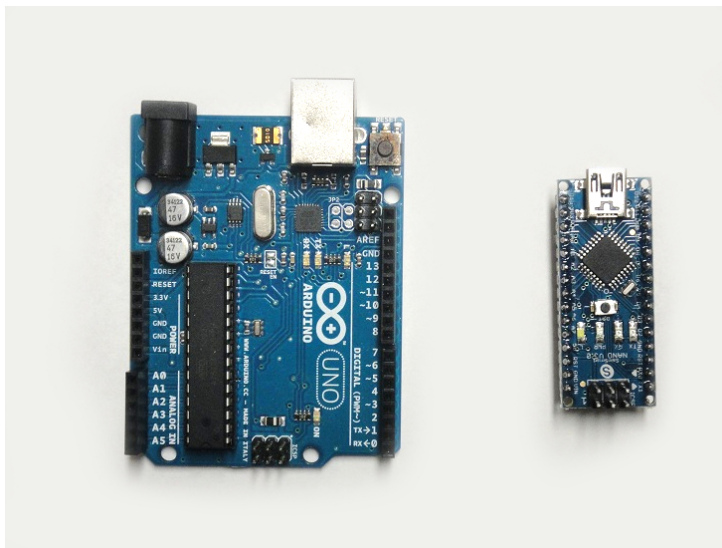
Arduino razvojno okolje (v nadaljevanju IDE) je bilo zasnovano na osnovi okolja programa Processing. Arduino je lahko programiran v jeziku C in C++. Zanje je zelo veliko zanimanje, obstaja tudi ogromno knjižnic, ki so zanj dostopne.

Za našo razvojno ploščico Arduino smo izbrali Arduino Nano (Slika 3.6). To je precej manjša in cenejša različica popularnejšega Arduino Uno. Zaradi manjšega faktorja pa ploščica ni nič manj zmogljiva. Njene specifikacije so naslednje [11]:

- Mikrokrmilnik Atmel ATmega328, ki imajo 14 digitalnih priključkov, od tega jih 6 omogoča PWM izhod. Ima tudi 8 analognih priključkov.

Vsebuje 32 KB flash pomnilnika, 2 KB SRAM-a in 1KB EEPROM-a za shranjevanje informacij, ki ostanejo po izključitvi mikrokontrolerja iz napetost.

- Arduino nano deluje na napetosti 5V, kot vhodno napetost pa lahko nanj priključimo med 6-20V, kar je primerno za napajanje ploščice preko baterije. Izhodni tok na priključkih je omejen na 40mA;
- Glavna odločitev za izbiro te ploščice so bile dimenzije le-te, ki znašajo približno 2cm x 4.5cm.



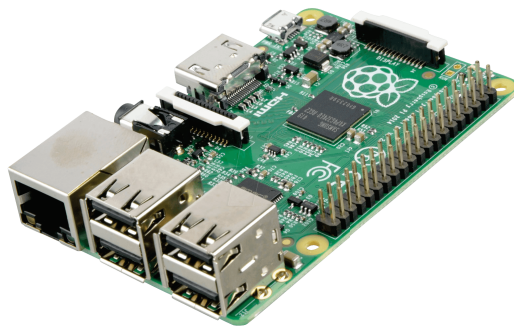
Slika 3.6: Prikaz sistemov Arduino uno na levi in Arduino nano na desni

3.3 Raspberry Pi računalniški sistem

Raspberry Pi je mikro računalnik (Slika 3.7), ki smo ga pri našem projektu uporabili prav zaradi njegovih malih dimenzij. V našem sistemu opravlja funkcijo glavnega modula. Namesto njega bi lahko uporabili katerikoli računalnik, ki ima nameščen operacijski sistem Windows ali Linux. Zaradi prenosnosti in nizke cene Raspberry Pi-ja pa smo se odločili zanj. Ker je na trgu na voljo že kar precej različic tega računalnika, smo se odločili za eno izmed prvih generacij Raspberry Pi, gre za model B, ki je izboljšana verzija prvega modela, ki je prišel na trg. Za ta model smo se odločili, ker za delovanje sistema ne potrebujemo visoke zmogljivosti, saj bo upravljal le manjše delo v našem projektu. Sprejemal bo rezultate iz senzorjev postavljenih po vrtnih površinah in jih pošiljal na strežnik. Tako je na nek način ta računalnik le zbirališče podatkov. Nekaj specifikacij o malem računalniku [12]:

- hitrost procesorja: 700MHz,
- velikost RAM-a: 512MB,
- ethernet priključek, 2 USB priključka, napajanje iz mikro USB priključka,

- operacijski sistem in podatki, naloženi na SD kartici,
- napajanje 5V in poraba približno 1A ob priključeni USB napravi,
- dimenzije: 85mm x 56mm x 17mm (D x Š x V).



Slika 3.7: Raspberry Pi

Vsak modul ima svojo identifikacijsko kodo, ki pomaga sistemu ločiti signale oz. določiti iz katerega modula prihajajo podatki. V našem projektu smo za identifikacijo uporabili kar cela števila (1, 2, ...). V splošnem pa bi lahko bila ta koda karkoli.

Poglavje 4

Delovanje sistema

V tem poglavju bomo podrobno opisali delovanje našega sistema. Na začetku bomo opisali kako v splošnem deluje sistem in njegove sestavne dele. Kasneje pa bomo podrobno pogledali še delovanje in zgradbo glavnih in oddaljenih modulov.

4.1 Osnovni gradniki sistema

Posplošen prikaz delovanja sistema je razviden na sliki 4.1. Zgradba sistema ima tri glavne komponente:

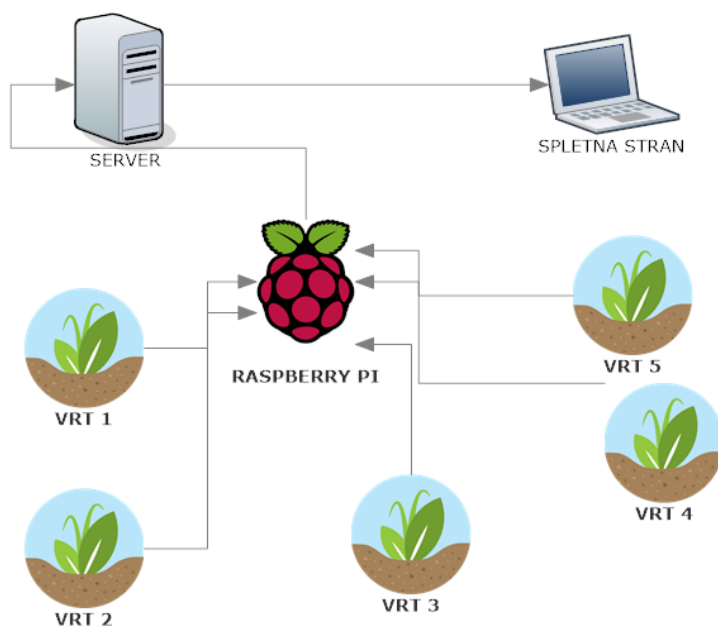
1. oddaljeni moduli,
2. glavni modul,
3. strežnik.

Oddaljeni moduli so namenjeni temu, da se postavijo na vrtno površine in brezžično glavnemu modulu pošiljajo podatke, ki jih beležijo s pomočjo vgrajenih senzorjev. Prav tako omogočajo samodejno zalivanje vrtov z vgrajenim relejem, ki preko ventila sproža dovod vode do namakalnih sistemov.

Glavni modul deluje na mini računalniku Raspberry Pi, na katerem teče programska koda, napisana v programskem jeziku Python. Ta sprejema po-

datke iz oddaljenih modulov. Nato pa jih posreduje naprej na strežnik. Podatke iz oddaljenih modulov pridobiva s pomočjo serijske komunikacije, saj je preko USB vodila povezan na Arduino-a, ki s pomočjo modula nRF24l01 komunicira z oddaljenimi moduli.

Tretji del predstavlja strežnik, na katerem je podatkovna baza, ki shranjuje podatke, katere dobiva iz glavnega modula. Na strežniku se nahaja tudi spletna stran, ki končnemu uporabniku na preprost način prikazuje relevantne podatke iz njegovih vrtov. Ti podatki, ki jih lahko spremlja od kjerkoli, mu omogočajo pregled nad trenutnim dogajanjem na njegovem vrtu. Ti uporabniku pomagajo pri odločitvah za možne izboljšave na vrtu.



Slika 4.1: Splošen prikaz delovanja sistema

4.2 Oddaljeni moduli

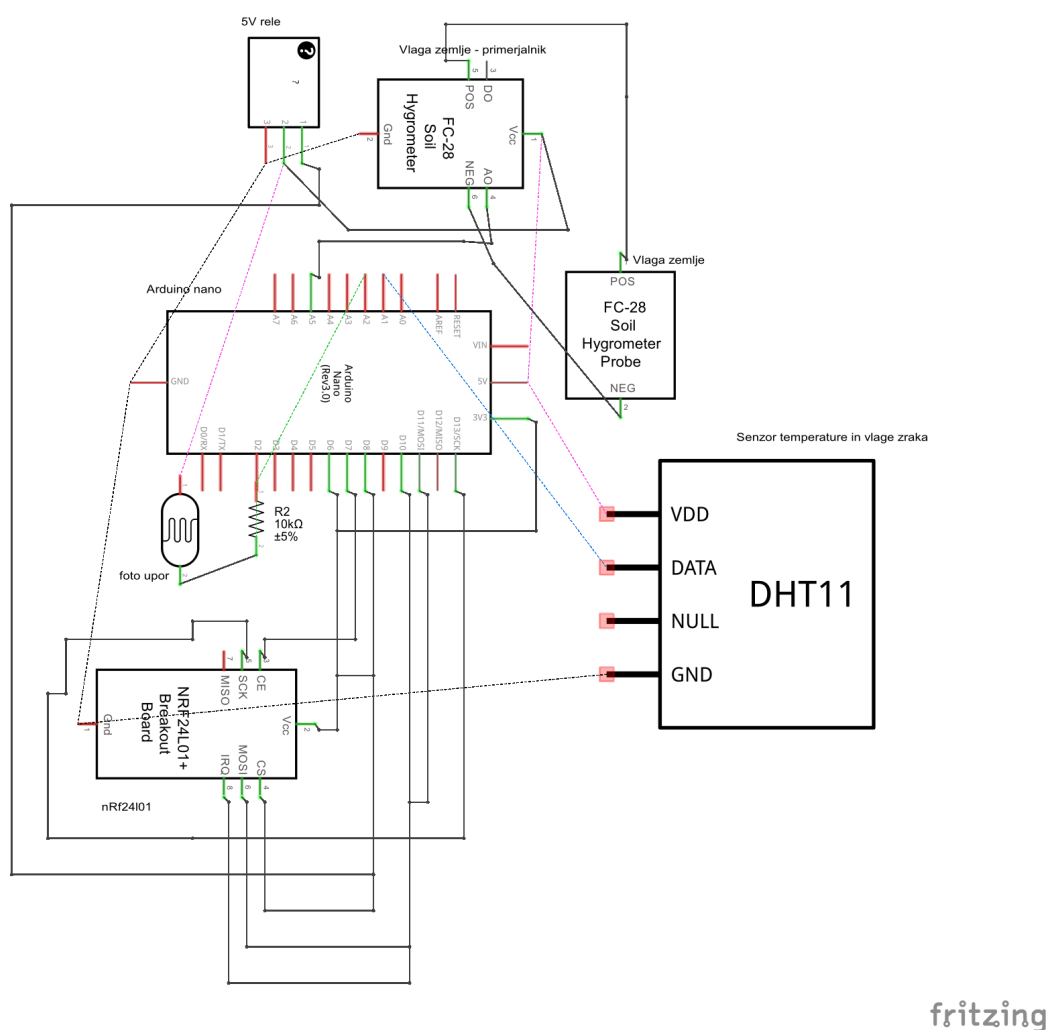
Oddaljeni moduli (Slika 4.2) delujejo na razvojni ploščici Arduino nano. Z njimi merimo štiri glavne dejavnike, od katerih je odvisen razvoj rastlin; poskrbimo pa tudi za njihovo mobilnost z možnostjo brezžične komunikacije. Tako so na oddaljene module priključeni naslednji gradniki:

- senzor FC-28 za merjenje vlage zemlje,
- senzor DHT-11 za merjenje vlage zraka in temperature okolice,
- foto upor za merjenje osvetljenosti,
- modul nRF24l01 za brezžično komunikacijo z glavnim modulom.

Slika 4.3 prikazuje dejansko vezavo oddaljenega modula po izdelavi. Razlog, da so senzorji pritrjeni na žice in ne na PCB, je potreba po njihovi premičnosti. Foto upor smo tako postavili na vrh vodoodpornega ohišja, s tem je bil omogočen direkten dostop do sončnih žarkov oz. drugih virov svetlobe. Podobno velja za senzor FC-28, ki smo ga lahko pritrdili na dno ohišja, ker ga je bilo potrebno zapičiti v zemljo.

```
1 int soilHum = A5;
2 int tempVrednosti[10];
3 int getSoilHumidity(){
4     for (int i = 0; i < 10; i++){
5         int vrednost=analogRead(soilHum);
6         int obrni=abs(analogRead((soilHum))-1024);
7         tempVrednosti[i]=obrni;
8         return povpreci(tempVrednosti)}
9 int povpreci(int vrednosti[i]){
10    int vsota=0;
11    for(int i = 0; i < 10; i++){
12        vsota+=vrednosti[i];
13    }
14    return vsota/10;
15 }}
```

Izvorna koda 4.1: Zajemanje podatkov s senzorja vlažnosti zemlje FC-28



Slika 4.2: Shema vezja oddaljenega modula

Senzor FC-28 je precej preprost za uporabo, saj je potrebno v programu le prebrati njegovo vrednost analognega priključka, na katerega je priključen. Podobno merimo tudi vrednosti iz foto upora. Oba parametra preberemo hitro, saj nimata velikih zamikov od spremembe dejanske vrednosti do spremembe, ko se to pozna na senzorju. V spodnji kodi je opisan postopek branja podatkov iz senzorja FC-28 (Izvorna koda 4.1). Podoben postopek je pri foto uporu. Pri senzorju FC-28 imamo le en dodaten korak, ki nam izračuna recipročno vrednost. Brez tega bi dobili manjšo vrednost pri bolj mokri zemlji



Slika 4.3: Izdelan oddaljeni modul s senzorji

in obratno. V obeh primerih pa je potrebno vzeti povprečje vrednosti, saj s tem dobimo bolj natančne podatke.

Za uporabo senzorja DHT-11 smo uporabili kar ustrezno knjižnico, saj so algoritmi za ta senzor precej kompleksni [6].

modul nRF24L01 je med najbolj kompleksnimi moduli v našem sistemu, zato smo tudi zanj uporabili knjižnico [13]. Uporaba knjižnice je precej preprosta. Ima pa veliko konfiguracij, ki jih je potrebno spremeniti, da lahko sistem sploh komunicira med moduli in da deluje karseda dobro. To pomeni, da ima čimvečji domet in čim večjo zanesljivost prenosa informacije. Med pomembnejšimi nastavitvami so bile:

- Nastavitev kanala komunikacije med dvema moduloma. Vsi naši moduli, tako glavni kot oddaljeni, se nahajajo v istem kanalu, tako da se

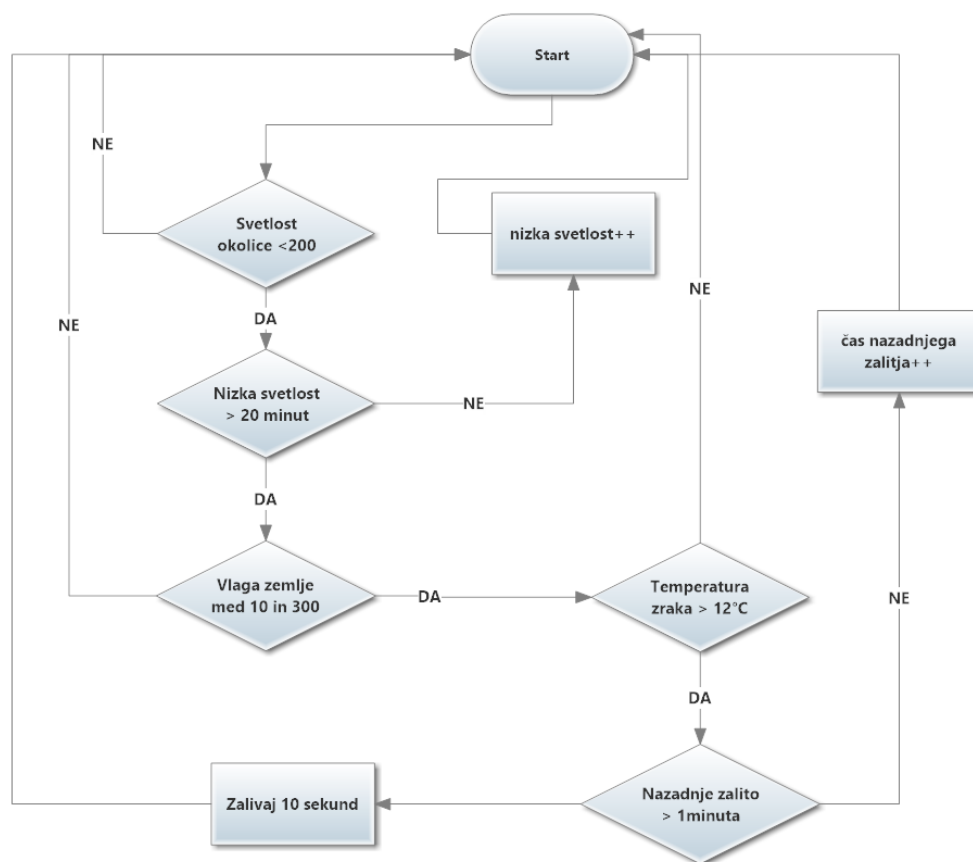
vsi slišijo med seboj. Kanal lahko nastavimo na kakršnokoli kombinacijo petih byte znakov, npr. "0A5A1".

- Nastavitev "AutoAck" predstavlja možnost pošljanja potrdila, da je modul pravilno sprejel paket. Tega nismo želeli, saj smo želeli zmanjšati količino prometa med moduli. Če pa bi se vmes kakšen paket izgubil, pa tudi ne bi bilo kritično.
- Nastavitev "setRetries" predstavlja nastavek, kolikokrat se paketek pošlje in časovni zamik med ponovnimi pošiljanji. Recimo, setRetries(10,15) nastavi, da se paketek pošlje 10-krat, zamik med njimi pa je 15 milisekund. To funkcijo smo optimizirali s testiranjem do čim manjše vrednosti, ko podatki še vedno prihajajo z zanesljivo in zadovoljivo hitrostjo [13].

4.2.1 Avtomatično namakanje zemlje

Naš oddaljen modul ima tudi možnost namakanja zemlje, ki se zgodi pod določenimi pogoji. Sistem deluje s pomočjo solenoidnega ventila. To je vodni ventil, ki ga lahko po želji vključimo ali izključimo z dovodom električnega toka. Ta ventil deluje na 12V napetost, zato je bila potrebna baterija s takšno napetostjo. Dovod napetosti, da se ventil odpre, se regulira s 5V relejem, ki smo ga lahko upravljali direktno iz Arduina-a. Na začetek in konec ventila smo namestili nastavek za povezavo na navadno vrtno cev, katere en konec gre na dovod vode, drugi konec cevi pa se priključi na nastavek za zalivanje, npr. škropilnik, porozna cev za namakanje. Kot alternativa solenoidnega ventila za manjše vrtove oziroma lončnice je priključitev na majhno vodno črpalko.

Delovanje sistema zalivanja je posplošeno prikazano na sliki 4.4. Bistvo delovanja sistema je, da se zalivanje vključi glede na trenutne okoljske pogoje. Zato je potrebno definirati časovne intervale zalivanja glede na določene parametre. Ker naš sistem pošlje podatke enkrat na sekundo, smo del kode, ki preveri, če se lahko začne zalivanje, postavili kar zraven in s tem dovolj



Slika 4.4: Posplošen prikaz algoritma za zalivanje

natančno merili časovne intervale, ki so pomembni v tem algoritmu. Preden se cikel zalivanja začne, algoritem na oddaljenem modulu preveri ali je osvetljenost okolice manjša od vrednosti 200, kar je naloga foto upora. S tem se prepreči, da bi se zalivanje sprožilo ob močnem soncu, kar bi poškodovalo ali celo uničilo rastline (učinek povečevalnega stekla sončnih žarkov na vodne kapljice). Ugotavljali smo, da je foto upor vrnil vrednost 200, ko je bila noč ali močna oblačnost. Da bi izključili možnost kratkotrajne nizke osvetljenosti, npr. prehodna osenčenost senzorja zaradi kakšnega objekta med njim in soncem, smo v naslednjem koraku preverili ali stanje nizke svetlobe traja vsaj 20 minut. Če pogoj 20 minut ni izpolnjen, povečamo vrednost števca, ki meri, kako dolgo je svetlost okolice pod 200.

Stanje	vrednosti senzorja
Senzor v zraku	0-8
Senzor v zelo suhi zemlji	30-150
Senzor v rahlo namočeni zemlji	150-250
Senzor v srednje namočeni zemlji	250-350
Senzor v zelo namočeni zemlji	350-700
Senzor v vodi	700-950

Tabela 4.1: Vrednosti senzorja pri različnih primerih uporabe senzorja

V nadaljevanju začnemo preverjati ali senzor za vlago zemlje vrača vrednost med 10 in 300. V tabeli 4.1 smo preizkusili korelacijo med razmočenostjo zemlje in vrednostjo, ki nam jo vrača senzor vlage zemlje. Za zgornjo vrednost smo si izbrali 300, saj predstavlja optimalno vrednost vlage v zemlji, t.j. ko zemlja ni suha in je namočena do meje, ko je voda enakomerno razporejena po zemlji brez bazenov vode. Spodnja vrednost 10 pa je le iz razloga, da ko senzor premaknemo iz zemlje, njegova vrednost pade pod 10. Če tega pogoja ne bi dodali, bi se zalivanje začelo ob vsaki prestavitvi oddaljenega modula. Nato sledi preverjanje temperature okolice. Meja 12°C je bila postavljena iz dveh razlogov. Rastline v fazi hibernacije v mrzlem delu leta ne potrebujejo večjih količin vode, zadostujeta le deževnica in stopljen sneg. Drugi razlog, da ne želimo vklopiti ventila v mrzlih dneh, še posebej, če temperature nihajo blizu 0°C , je možnost zamrznitve vode v cevi, s čimer bi se poškodovali tudi deli našega sistema.

Zadnji pogoj, ki mora biti izpolnjen pred ponovnim zalivanjem je, da smo nazadnje zalivali pred približno eno minuto. S tem omogočimo, da se voda enakomerno razporedi po zemlji in zagotovimo, da bodo podatki iz senzorja vlage zemlje bolj natančni. Če je pogoj ene minute izpolnjen, se prične zalivanje, ki traja 10 sekund. Nato se cikel ponovi. V primeru da bodo pogoji v naslednji minuti spet izpolnjeni, se bo zalivanje nadaljevalo. Takšen način zalivanja je počasen, je pa blizu naravnemu in s tem bolj zdrav za rastline. Programska koda za zalivanje je prikazana v 4.2.

```
1 int casTeme = 0, nazadnjeZalito=0, casZalivanja = 0;
2 boolean zalivam = false;
3 void preveriZalivanje() {
4     if(getLight() < 200){
5         casTeme++; }
6     else casTeme=0;
7     if (zalivam) {
8         casZalivanja++;
9         if(casZalivanja<10){
10             ustaviZalivanje();
11             casZalivanja=0; } }
12     else if (!zalivam) {
13         if (getLight() < 200 && casTeme > 1200 ) {
14             if (getSoilHumidity() > 10 && getSoilHumidity() < 300) {
15                 if (getTemperature() > 12) {
16                     if (nazadnjeZalito > 60) {
17                         zalivam = true;
18                         nazadnjeZalito = 0;
19                         zacniZalivanje(); }
20                 else {
21                     nazadnjeZalito++;
22                     }}}} }
```

Izvirna koda 4.2: Algoritem zalivanje v programski kodi

4.3 Glavni modul

Glavni modul se sestoji iz dveh delov. Prvi je Arduino nano, ki je z USB kablom povezan z Raspberry Pi. Naloga tega dela je, da pridobiva podatke od drugih oddaljenih modulov oz. podatke iz senzorjev. Povezava poteka preko nRF24l01 modula. Brežžični moduli nRF na glavnem in na oddaljenih modulih so na istem kanalu. Med seboj komunicirajo samo moduli, ki so na istem kanalu, kar v praksi pomeni, da lahko naredimo sisteme, ki med seboj ne bodo mogli komunicirati. To je uporabno v primeru, ko je več sistemov blizu, saj s tem onemogočimo, da bi podatki enega sistema prehajali v drug sistem. Na ta način jih torej izoliramo med seboj. Glavni modul sprejema

podatke vseh modulov na enak način.

V našem projektu smo se odločili za eksterni del (Arduino del), ker smo želeli, da lahko uporabnik namesto Raspberry Pi-ja uporabi katerokoli platformo. Zato tudi modul nRF24l01 ni priključen direktno na Raspberry Pi. Edina razlika je program, ki teče na delu, ki sprejema podatke preko Arduina. Glavni modul ima poleg brezžičnega modula nRF24l01 tudi dva senzorja. Eden je DHT-11, ki odčitava temperaturo in vlago v zraku. Drugi senzor je BMP085, ki odčitava zračni pritisk, tudi za ta senzor smo uporabili knjižnico [9]. Na sliki 4.5 je narisana shema vezja, na sliki 4.6 izdelano vezje. Na glavnem modulu smo lahko senzorje namestili samo na PCB, saj ni potrebe, da bi bili nameščeni na različnih mestih, kot je to potrebno pri oddaljenih modulih.

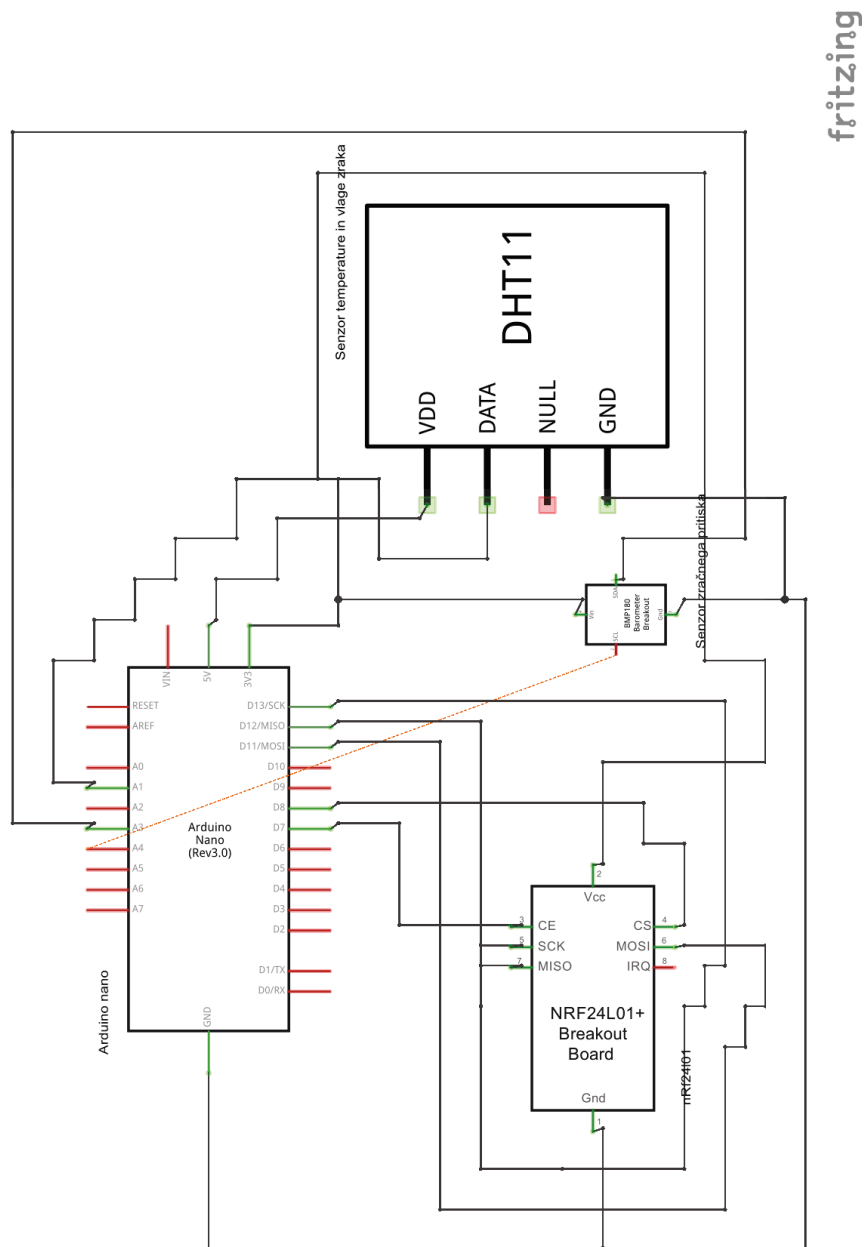
Drugi sestavni del glavnega modula je v našem primeru Raspberry Pi. Njegova naloga je, da podatke, ki jih prejme preko USB kabla, pošlje na strežnik. Razlikuje dve vrsti podatkov - podatke, ki jih pridobiva iz senzorjev priključenega Arduina in podatke, ki jih sprejema iz oddaljenih modulov. Podatki se razlikujejo tako, da imajo oddaljeni moduli, ki beležijo osvetljenost okolice, vlažnost zemlje ter temperaturo in vlago zraka, identifikacijsko številko od 2 in naprej. Senzorji iz priključenega Arduina beležijo le temperaturo, vlago in zračni pritisk, identificirajo pa se s številko 1. Ti podatki so informativne narave, saj nam sporočajo razmere v okolici glavnega modula in dajejo referenčno točko razmer na glavnem in oddaljenih modulih.

Program, ki teče na Raspberry Pi, je napisan v Python programskem jeziku. Deluje tako, da se ob zagonu najprej inicializira serijska komunikacija, nato pa se začne sprejemanje in pošiljanje podatkov v strežnik. Najprej se podatki v programu preberejo, nato se razdelijo v tabelo, saj so podatki, ki jih dobi, ločeni z znakom ';'. V nadaljevanju program preveri identifikacijo, ki je zapisana v dobljenih podatkih. Če identifikacijska koda izhaja iz priključenega Arduina, program zapiše parametre senzorjev na način, kot je definiran v podatkovni bazi za glavni modul. Če pa je identifikacijska koda večja od 1, jih zapiše na definiran način v bazi za oddaljene module. Nato

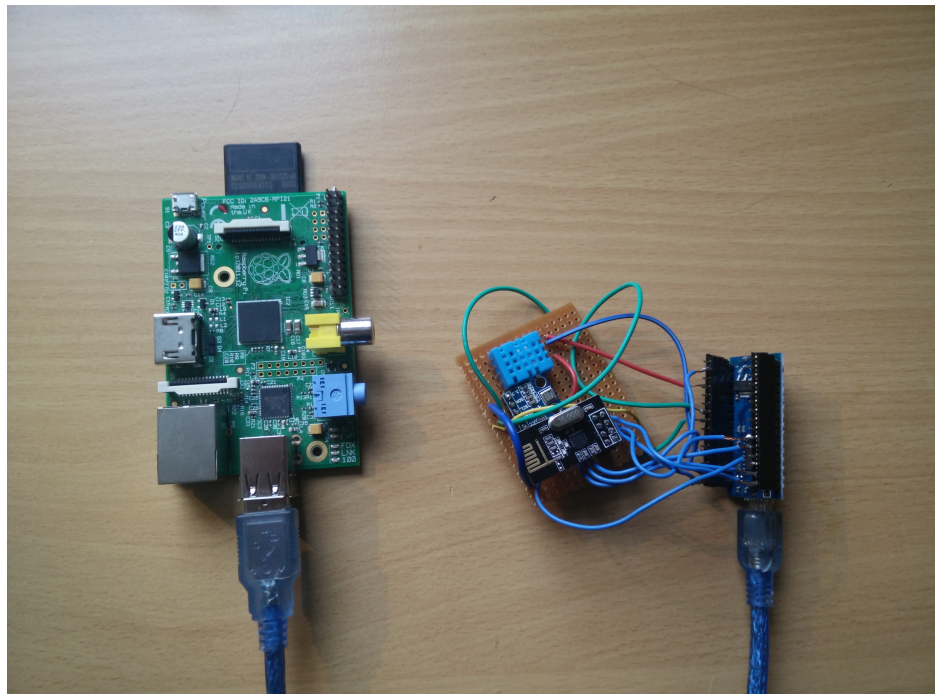
se podatki pošljejo na strežnik. Programska koda, ki teče na Raspberry Pi je prikazana v 4.3.

```
1 import serial, urllib
2 ser = serial.Serial(
3     port='/dev/ttyUSB0',
4     baudrate = 9600,
5     parity=serial.PARITY_NONE,
6     stopbits=serial.STOPBITS_ONE,
7     bytesize=serial.EIGHTBITS,
8     timeout=1)
9 counter=0
10
11 while 1:
12     x=ser.readline()
13     if (len(x)>0):
14         razdeljeno=x.split(";")
15         razdeljeno[-1]=razdeljeno[-1].strip()
16         if (razdeljeno[0] is "1"):
17             params=urllib.urlencode({"SensorID": razdeljeno[0],
18                                     "AirPressure": razdeljeno[1], "AirHumidity": razdeljeno[2],
19                                     "Temperature": razdeljeno[3]})
20         else:
21             params=urllib.urlencode({"SensorID": razdeljeno[0],
22                                     "Luminosity": razdeljeno[1],
23                                     "AirHumidity": razdeljeno[2], "Temperature": razdeljeno[3],
24                                     "SoilHumidity": razdeljeno[4]})
25
26 f = urllib.urlopen("http://gardencloud.azurewebsites.net/api/
    data", params)
```

Izvorna koda 4.3: Program za pošiljanje podatkov na strežnik



Slika 4.5: Vezalna shema glavnega modula



Slika 4.6: Izdelan glavni modul

Poglavje 5

Postavitev strežnika in spletne strani

Ko smo izdelali oddaljene module, glavni modul in vzpostavili povezavo med njimi, smo se lotili izdelave strežnika in spletne strani. Uporabili smo precej različnih orodij za izdelavo različnih elementov. Strežnik in spletna stran tečeta kar na Azure platformi [16], ki je zelo prilagodljiva glede izbire cene, saj nismo potrebovali veliko prostora, niti hitrosti povezave. Ta del predstavlja oblak v našem projektu.

5.1 Strežnik

Glavna funkcija našega strežnika je pridobivanje podatkov iz oddaljenih modulov, jih shraniti v podatkovno tabelo in nato posredovati spletni strani, ki jih prikaže. Naš strežnik temelji na ASP.NET platformi [14], programski jezik, ki smo ga uporabljali je C#, za programersko okolje smo uporabili Microsoft-ov program Visual Studio. Preko tega okolja smo tudi naložili strežnik na Azure. Paket, ki smo ga izbrali na Azure, vsebuje 2GB prostora, kar je več kot dovolj za naše podatke. Kopičijo se namreč le podatki o trenutnem stanju modulov. Ko smo testirali naš strežnik, smo v Visual Studiu izbrali le možnost Publish in strežnik je že deloval na spletni domeni, ki je v

našem projektu bila <http://gardencloud.azurewebsites.net> [15].

ASP.NET je odprtokodno orodje za izdelavo strežnikov. Namenjeno je izdelavi dinamičnih spletnih aplikacij, spletnih strani in spletnih storitev. Sestavni deli pri izdelavi spletnih aplikacij so Web Forms (spletni obrazci) [14].

5.1.1 REST srežnik

Naš strežnik je REST strežnik. REST je programska arhitektura pri izdelavi aplikacij, ki za svojo uporabo potrebujejo dostop do spleta. REST nam omogoča komunikacijo med glavnim modulom in strežnikom s pomočjo protokola HTTP. Pri povezavi glavnega modula in strežnika smo uporabili POST ukaz, ki se izvede na Raspberry Pi-ju in posreduje podatke, ki jih je pridobil iz oddaljenih senzorjev ali od svojega priključenega Arduina strežniku. V POST-u je potrebno navesti spletni naslov do strežnika oz. spletni naslov na strežniku, ki lahko sprejme POST ukaz s parametri, ki smo jih dodali temu ukazu. V našem primeru so bili parametri različni podatki iz senzorjev.

Strežnik sprejme te podatke in jih shrani v podatkovno bazo. Uporabljamo tudi ukaz GET, ki ga zahteva naša spletna stran. Ta ukaz v našem primeru pridobi zadnjih nekaj podatkov, ki jih ima shranjena podatkovna tabela in jih za oddaljene module prikaže v obliki grafa in geografske lokacije. Če pa pridobi podatke iz glavnega modula, jih prikaže na vrhu strani. Uporabljamo še en POST ukaz iz spletne strani na strežnik, ki se sproži, če uredimo določene podatke na spletni strani, kot so lokacija oddaljenega modula, ime oddaljenega modula, elektronski naslov za obveščanje uporabnika.

5.1.2 SQL podatkovna baza

Za izdelavo podatkovne baze smo uporabili Microsoft SQL, za načrtovanje te baze pa okolje SQL server. Naša podatkovna baza shranjuje dve tabeli. Prva tabela je tabela modulov in ima naslednje attribute:

- ID - shranjuje identifikacijsko številko modula,

- name - shranjuje ime modula. V projektu smo module poimenovali glede na to, kje je bil senzor postavljen. Npr. senzor na vrtu s paradizniki smo poimenovali Potato garden,
- coordinates - shranjuje geografske koordinate modula. Te podatke lahko uredimo v spletni strani. Z njimi, za lažji pregled nad vrtovi, določimo kje je naš vrt lociran,
- e-mail - shranjuje elektronski naslov tistega, ki se ga obvešča, če je zemlja v vrtu presuha. Ta atribut prav tako nastavimo preko spletne strani.

Druga tabela shranjuje attribute, ki predstavljajo vrednosti senzorjev. Obe tabeli sta povezani s primarnim ključem, tako da lahko ločimo podatke senzorjev glede na njegov ID. Njeni atributi so:

- ID - identifikacija, ki nam pove, od katerega modula shranjujemo podatke,
- humidity - shranjuje trenutno vrednost vlage v zraku,
- soilhumidity - shranjuje trenutne podatke o vlažnosti zemlje,
- temperature - shranjuje trenutno temperaturo zraka,
- airpressure - shranjuje trenutni zračni pritisk,
- luminosity - shranjuje trenutno osončenost vrta,
- date - shrani, kdaj so bili podatki shranjeni v tabelo, s tem pa jih lahko kasneje grafično. prikažemo.

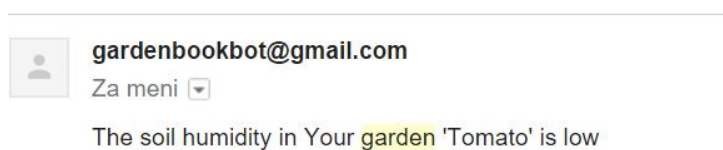
Ker oddaljeni modul in glavni modul ne pošiljata enakih podatkov od senzorjev, se v tabelo zapiše vrednost -1, če modul ne pošilja tega podatka. Na spletni strani pa ločimo podatke glavnega modula od ostalih tako, da za pridobitev njegovih podatkov vedno kličemo podatke iz identifikacijske številke 1.

5.1.3 Obveščanje uporabnika po elektronski pošti

V naš projekt smo dodali še elektronsko obveščanje skrbnika vrta v primeru, ko vlažnost zemlje pade pod določeno kritično vrednost. To se lahko zgodi v treh primerih:

- možnost mehanske napake - ni dovoda vode, ker se le-ta nekje v sistemu prekine,
- nepovezanost oddaljenega modula z ventilom, ki sprošča dovod vode ali pa se ventil pokvari,
- izsušitev zemlje, kadar ni vode v vodovodnem sistemu.

Sistem obveščanja se sproži, kadar senzor pokaže vrednost med 50 in 10. To pomeni, da je zemlja zelo suha. Ne more pa se sprožiti, če pade vrednost pod 10, ker to z veliko verjetnostjo pomeni, da je bil modul iztaknjen iz zemlje. Ko dobimo podatek iz modula, da je na vrtu zemlja suha, se na elektronski naslov pošlje sporočilo, da je na točno določenem vrtu (npr. tomato garden) močno padla vlažnost zemlje. To elektronsko sporočilo se pošilja na največ triurne intervale, da ne poplavimo elektronskega nabiralnika. Elektronski naslov je lahko drugačen za vsak vrt posebej, kar je uporabno, če imajo vrtovi različne oskrbovalce, saj s tem vsakemu posebej pošlje sporočilo samo za njegov vrt. Primer sporočila je na sliki 5.1.



Slika 5.1: Primer elektronskega sporočila

5.2 Spletna stran

Spletna stran je namenjena prezentaciji podatkov iz vrtov uporabniku. Želeli smo, da je spletna stran:

- preprosta za uporabo,
- pregledna in nenasičena,
- daje uporabniku le pomembne podatke,
- uporabna tudi na mobilnih napravah.

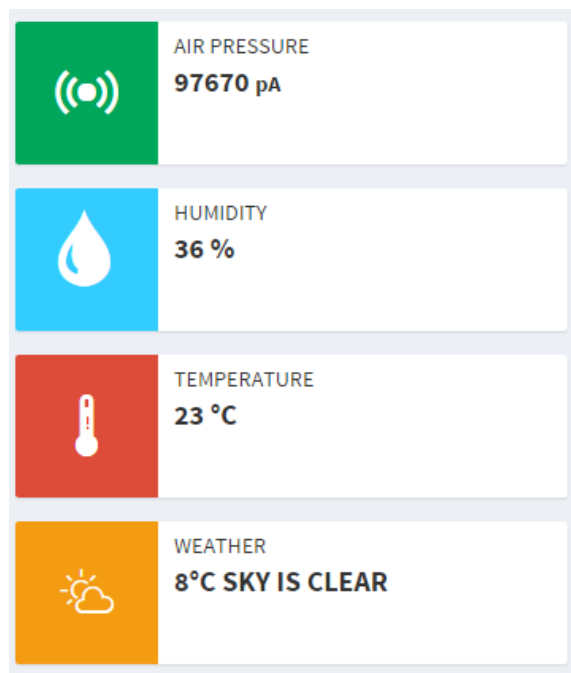
Kot že omenjeno, naša spletna stran pridobiva podatke iz modulov preko GET ukaza iz strežnika. Ta ukaz se sproži ob zagonu spletne strani. Od podatkov iz glavnega modula spletna stran uporabi le zadnji vnos v podatkovno tabelo. To dobimo s pomočjo SQL ukaza TOP in jih sortiramo po datumu. Ti podatki so le informativne narave. Iz oddaljenih modulov se uporabi za grafični prikaz le nekaj zadnjih podatkov (ob zagonu spletne strani zadnjih 500 podatkov), število let lahko določimo s pomočjo znaka + in - ob grafu ter s tem spreminjamo časovni razpon na grafu. Preberejo se še podatki kot so koordinate in ime vrta.

Spletna stran je narejena v AngularJS, ki je odprtokodno ogrodje za izdelavo dinamičnih spletnih strani [17]. V našem projektu smo uporabili precej knjižnic in en API za pridobivanje podatkov. API je namenjen pridobivanju trenutnih vremenskih razmer. API deluje tako, da spletni naslov spremenimo na naše koordinate vrta. Ta pa nam nazaj v obliki XML sporoča trenutne razmere. V našem projektu uporabimo trenutno temperaturo, ki jo lahko primerjamo z izmerjeno s strani naših modulov in trenutni opis vremenskih razmer, npr. oblačno, megleno, ...

Naša spletna stran je v času pisanja tega projekta objavljena na spletnem naslovu <http://gardencloud.azurewebsites.net/>.

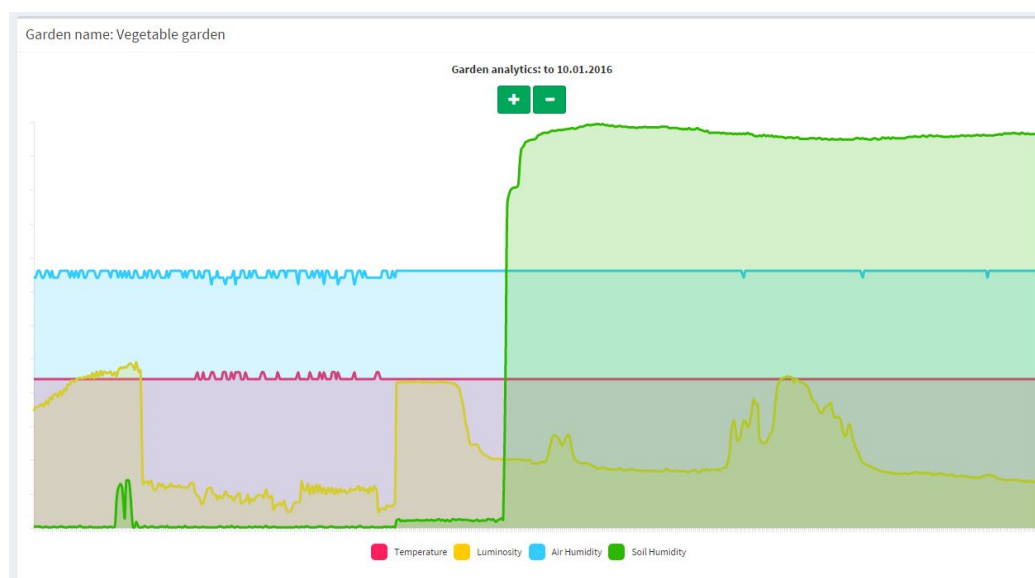
Sestavni deli naše spletne strani so prikazani v nadaljevanju.

- Podatki iz glavnega modula - zračni pritisk, zračna vlaga, temperatura in na koncu še trenutna temperatura in opis vremena, ki je vezan na trenutni vrt. Te podatke pridobivamo iz spleta. Slika 5.2 prikazuje prikaz teh podatkov na spletni strani.



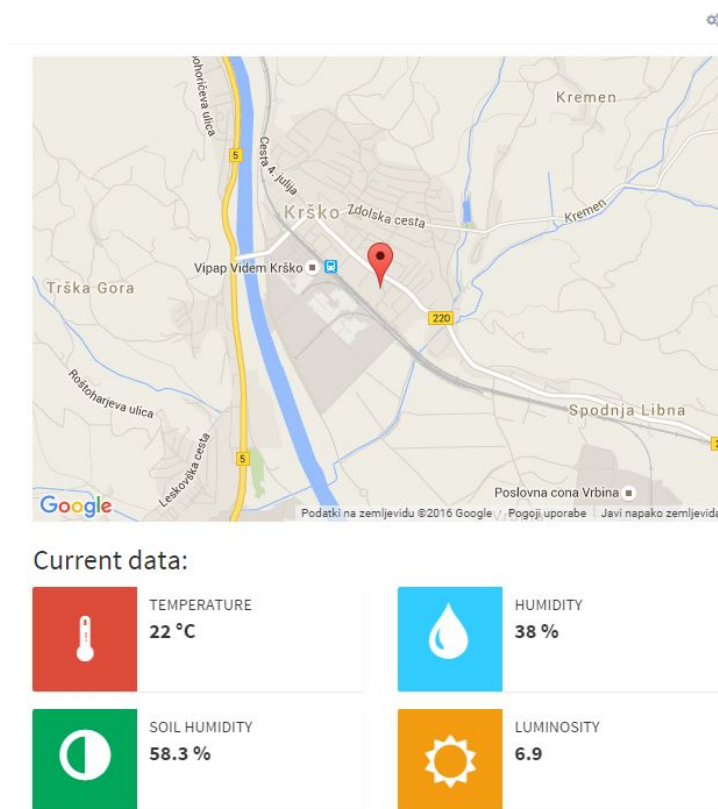
Slika 5.2: Spletna stran s podatki glavnega modula in trenutne vremenske razmere

- Na spletni strani se nahaja tudi graf, ki prikazuje spremembo podatkov iz senzorjev v odvisnosti od časa. Podatki so pobarvani logično, da uporabnik hitro najde podatke, ki ga zanimajo. Poleg tega pa mu pri tem pomaga še legenda podatkov. Uporabnik ima na voljo še dva gumba + in - za večanje in manjšanje časovnega obsega podatkov. Prikaz grafa na sliki 5.3.



Slika 5.3: Spletna stran z grafom poteka okoljskih parametrov

- Desno od grafa je prikazana lokacija vrta na zemljevidu. Pod zemljevidom se nahajajo trenutni oz. zadnji podatki iz grafa, tako uporabnik hitro izve trenutne razmere na vrtu. Izgled teh gradnikov je prikazan na sliki 5.4.
- Na dnu spletne strani je prikaz najvišjih in najnižjih vrednosti podatkov vseh senzorjev. Podatki se spreminjajo glede na trenutni graf. Če spreminjamo časovni interval v grafu, se tudi ti podatki spreminjajo. Namenjeni so uporabniku, ki lahko spremlja največje spremembe skozi dan na svojem vrtu. Prikazano na sliki 5.5.
- Podatki, kot so ime vrta, lokacija vrta in elektronski naslov, na katerega se pošlje opozorilo o presuhem vrtu, se nastavijo v okenčku, do katerega pridemo preko gumba v obliki zobnika nad zemljevidom. Prikazano na sliki 5.6.



Slika 5.4: Zemljevid lege in trenutne vrednosti okoljskih parametrov vrta



Slika 5.5: Najvišje in najnižje vrednosti podatkov

Edit info

Garden Name

Vegetable garden

GPS Coordinates

45.9550826

15.4971353

Email notification to:

luka.colaric@gmail.com

Close

Save changes

Slika 5.6: Vnos in sprememba podatkov

Poglavje 6

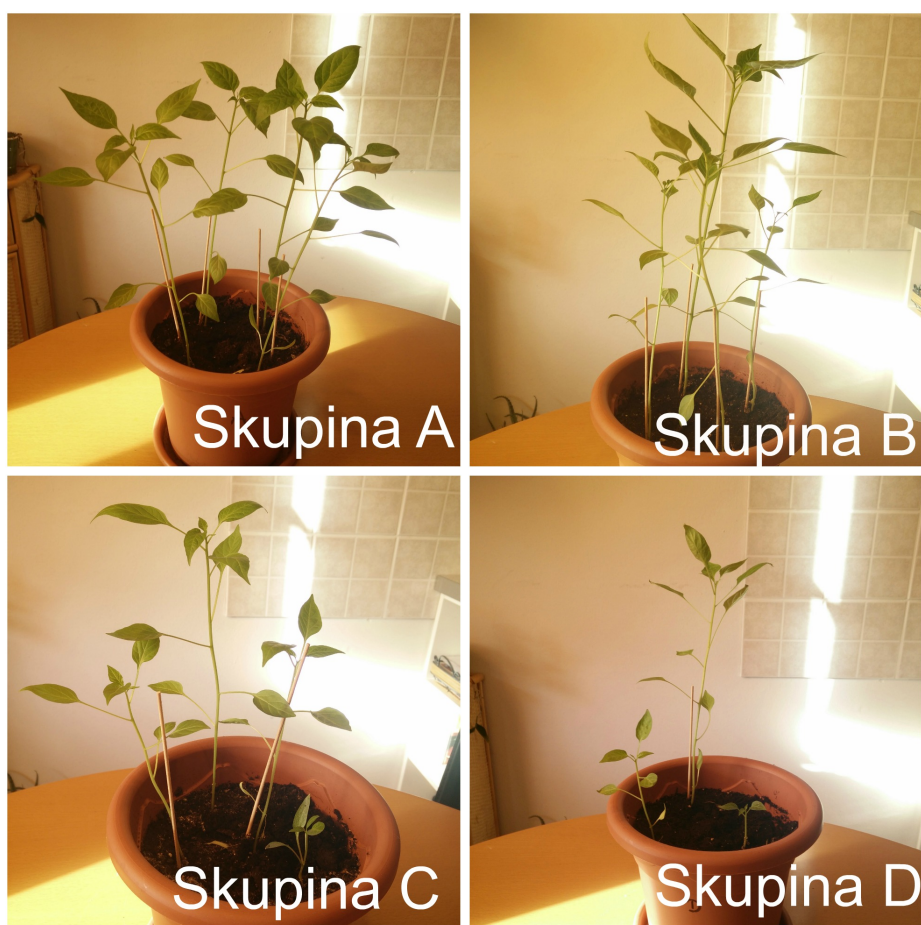
Preizkus sistema v praksi

V tem poglavju bomo opisali testiranje našega sistema. Prav tako so predstavljeni rezultati, ki smo jih preko testiranja pridobili. Testiranje je potekalo na pravih rastlinah, kot bi se to dogajalo v realni uporabi našega sistema. Testno okolje naj bi bil naš vikend, vendar zaradi zime vseh testov nismo mogli vseh testov opraviti tam. Zato smo na vikendu opravili le test dometa. Test, kako naš sistem deluje na rast rastlin pa smo izvedli kar doma. Tu smo ga najlažje nadzirali.

6.1 Primerjalni preizkus različnih načinov zalivanja

En del projekta smo testirali na čiliju, vrsti paprike, ki je občutljiva rastlina in za zdrav razvoj zahteva zelo ugodne pogoje. Odločili smo se za vrsto Carolina Reaper [18], ki je v času izdelave tega projekta najmočnejši čili na svetu. Semena smo sprva posadili v skupen lonček, ko pa so rastlinice razvile po dva para listkov, smo jih po štiri do pet razdelili v štiri različne testne lončke. Vsi lončki so imeli isti vir svetlobe in čas osvetlitve - žarnica, ki je gorela 12 ur na dan, s čimer smo simulirali idealno osvetlitev, ki jo ta vrsta rastline potrebuje. Lončki so se med seboj razlikovali le glede na režim namakanja. Postavitev lončkov je prikazana na sliki 6.2. Postavili

smo jih na domačo mizo in jih tako imeli ves čas pod nadzorom. V času predstavitve rezultatov so bile rastline stare približno dva meseca. Celoten proces zalivanja pa je trajal približno dva meseca in pol. Lončke smo glede na režim namakanja razvrstili v štiri skupine. Na sliki 6.1 so prikazani vsi testni lončki.



Slika 6.1: Primerjava vseh testnih lončkov

6.1.1 Rezultati preizkusa zalivanja



Slika 6.2: Sobni testni prostor zalivalnega sistema našega projekta

1. skupina A - namakanje je bilo samodejno s pomočjo našega algoritma zalivanja,
2. skupina B - ročno zalivanje glede na podatke, ki jih je uporabnik izvedel iz naše spletne strani,
3. skupina C - ročno zalivanje vsak tretji dan ob isti uri,
4. skupina D - ročno zalivanje "po občutku".

Ker smo želeli testirati le pomen režima namakanja na rast in razvoj rastlin, smo glede ostalih pomembnih parametrov morali zagotoviti čim bolj podobne pogoje.

- Podobne začetne rastline: za presaditev v testne lončke smo izbrali rastlinice, ki so si bile čim bolj podobne po velikosti in zdravju.

- Svetloba: vsi testni lončki so bili izpostavljeni istemu viru svetlobe (naravna svetloba in žarnica) in času osvetlitve (12 ur na dan - od 7. do 19. ure).
- Izpostavljenost svetlobi: zaradi možnega zakrivanja rastlin med seboj, smo lončke vsak dan obrnili za približno četrtno kroga in s tem dosegli, da so bile v povprečju vse rastline zelo podobno izpostavljene svetlobi.
- Temperatura: ker čiliji za zdrav razvoj nujno potrebujejo toploto, smo jo najbolj enostavno kontrolirali z radiatorjem. Toplota čez dan je bila približno 25°C, čez noč okoli 20°C.
- gnojenje: vse rastline smo gnojili enako in sicer enkrat na dva in pol tedna.



Slika 6.3: Končna izvedba testnega namakalnega sistema s peristaltično črpalko

6.1.2 Skupina A

V skupini A so bile posajene štiri rastline, ki so bile avtomatično namakane z našim algoritmom namakanja, s pomočjo peristaltične črpalke. Sistem za to

6.1. PRIMERJALNI PREIZKUS RAZLIČNIH NAČINOV ZALIVANJA⁴⁵

skupino je bil postavljen tako, da smo ob presaditvi rastlin v testni lonček, zraven namestili še naš oddaljeni modul. Ta modul je opravljal le funkcijo zalivanja in ni bil namenjen pošiljanju trenutnih podatkov v našo podatkovno bazo.

Na modul smo namestili rele in peristaltično vodno črpalko. Ta vrsta črpalke je bila idealna za naše male lončke, saj ima pretok vode približno 100 ml na minuto, kar je zadovoljivo namočilo zemljo v njem. Prikazano na sliki 6.3. Namestili smo jo tako, da smo en konec črpalke namočili v velik kozarec vode, drugi konec pa v zemljo na sredino lončka. Čeprav je bil na sredini lončka tudi naš senzor, se je voda po nekem času dovolj dobro razporedila in enakomerno namočila zemljo. Izgled rastlin skupine A po testnem obdobju je prikazan na sliki 6.4. Ugotovitve:



Slika 6.4: Testni lonček A

- Vse štiri rastline so razvile podobno velikost, kar nam sporoča, da je bila voda enakomerno razporejena po celotnem lončku.
- Vse rastline so zdrave, kar nam pove, da so bili pogoji za rast dobri. Zemlja ni bila nikoli preveč razmočena in nikoli presuha.
- S to skupino smo bili zelo zadovoljni, saj nam je v praksi pokazala, da je naš algoritem zalivanja pravilno in dobro deloval.

6.1.3 Skupina B

V testnem lončku B smo imeli prav tako posajene štiri rastline. Zalivali smo jih glede na vrednosti, ki jih je vračal oddaljeni modul na spletno stran. Dejansko je bila razlika glede na skupino A le, da smo zalivali sami. To pa ni bilo tako dosledno kot pri avtomatičnem zalivanju skupine A. Včasih smo pozabili zaliti rastline, včasih smo jih zalili preveč, čeprav vrednost razmočenosti zemlje še ni padla pod 300 (meja pri avtomatskem zalivanju). Če je bila npr. vrednost razmočenosti pozno zvečer še rahlo nad 300, smo zalili zato, da se preko noči zemlja ne bi preveč izsušila. S to skupino smo želeli ponazoriti, da lahko naše oddaljene module uporabljamo tudi brez vodne črpalke, samo kot referenco, kdaj bi bilo zalivanje optimalno. Izgled rastlin skupine B je prikazan na sliki 6.5. Ugotovitve:

- Razvile so se vse štiri rastline, vendar so bile dokaj različne glede višine. Ena izmed njih je zelo izstopala, ostale so bile precej nižje, a še vedno nekoliko manjše kot rastline v skupini A.
- Večina rastlin iz te skupine je bila zdravega izgleda, le najvišja je, zaradi hitre rasti, imela tanko in krhko steblo.
- Predvidevamo, da je do opisane razlike prišlo iz dveh razlogov. Po eni strani ročno zalivanje lahko privede do neenakomerne navlaženosti zemlje. Tako smo včasih zalili vse rastline enako, včasih pa je bila kakšna bolj, druga manj zalita. Po drugi strani pa je ročno zalivanje manj optimalno glede na dejansko vlažnost zemlje, kot če je avtomatsko.

6.1. PRIMERJALNI PREIZKUS RAZLIČNIH NAČINOV ZALIVANJA⁴⁷

- Vendar smo tudi s temi rezultati bili zelo zadovoljni, saj so nam sporočali, da sistem dobro deluje tudi, če nas samo opozarja, kdaj moramo zalivati rastline.



Slika 6.5: Testni lonček B

6.1.4 Skupina C



Slika 6.6: Testni lonček C

Ta skupina štirih rastlin je bila zalita periodično točno na tri dni z vedno enako količino vode, ne glede na vlažnost zemlje. S tem smo poizkušali ponazoriti namakalne sisteme, ki delujejo samo na čas in ne na dejanske razmere oz. potrebe. Kljub temu, da smo imeli naše testne rastline v prostoru, kjer ni bilo vpliva zunanjih dejavnikov, kot je dež, je bila namočenost zemlje vsake tri dni vedno malo drugačna. Glede na vlago v zraku, ki je bila iz dneva v dan drugačna, se je namreč spreminjalo tudi izhlapevanje vode iz zemlje. Namakali nismo do nobene določene vrednosti, tako kot tudi ne časovno od-

6.1. PRIMERJALNI PREIZKUS RAZLIČNIH NAČINOV ZALIVANJA⁴⁹

visni namakalni sistemi. Izgled rastlin skupine C je prikazan na sliki 6.6. Ugotovitve:

- Razvile so se vse štiri rastline, vendar precej različno. Ena rastlina je na robu preživetja, dve rasteta zelo počasi. Le ena od štirih rastlin je dokaj primerne rasti.
- Samo dve od štirih rastlin izgledata zdravi. Tretja ima rumene listke, četrta je izrazito zakrnela.
- Vzrok izraziti raznolikosti v izgledu rastlin je verjetno le močno nihanje namočenosti zemlje, ki se je lahko v treh dneh popolnoma izsušila ali pa je bila še zelo vlažna, ko je bil čas ponovnega zalivanja. Torej rastline v bistvu nikoli niso bile primerno zalite.
- Ta test nam je nazorno pokazal, da avtomatično časovno namakanje zelo slabo zadosti potrebam rastline po vodi.

6.1.5 Skupina D

Zadnja skupina petih rastlin je temeljila na preprostem sistemu zalivanja po občutku. Rastline je zalivala druga oseba, ki ni videla trenutnih podatkov iz skupine B, saj nismo želeli, da bi s tem spreminjala svoje običajne navade zalivanja. Osebi je predstavljal kriterij za zalivanje le izsušenost zemlje na površini. Režim zalivanja je bil tako popolnoma subjektiven, časovni intervali in količina vode različni. Rastline so bile zalite enkrat na dva dni do enkrat na teden. S tem testom smo želeli ponazoriti učinek preprostega zalivanja, kot ga ljudje v glavnem uporabljamo. Izgled rastlin skupine D je prikazan na sliki 6.7. Ugotovitve:

- V tej testni skupini sta od posajenih petih rastlin dve propadli. Preostale tri so se v višino razvile zelo različno.
- Preostale rastline izgledajo zdrave, vendar so v povprečju precej nižje kot v testih A in B. Le ena od rastlin je dosegla podobno velikost kot v prvih dveh skupinah.



Slika 6.7: Testni lonček D

- Vzrokov za takšen rezultat je več - neenakomerno zalivanje, različni časovni intervali, včasih je oseba na rastline pozabila, včasih pa jih je premočno zalila.
- S tem testom smo zelo dobro ponazorili običajno zalivanje rastlin. Tudi rezultati so temu primerni.

6.2 Testiranje dometa oddaljenih modulov

Pri našem projektu je pomemben tudi domet od oddaljenih modulov do glavnega modula. Domet smo testirali s pomočjo preprostega testa. Najprej smo prižgali glavni modul, da je začel sprejemati signal oddaljenih modulov. Nato smo prižgali oddaljen modul. Ko je glavni modul sprejel signal, je to izpisal v terminal. Ko smo se začeli oddaljevati, smo spremljali izpis podatkov na terminal glavnega modula preko SSH-ja. Povezali smo se preko zunanjega IP naslova glavnega modula. Na neki meji je signal prenehal prihajati na glavni modul, tako smo lahko določili meje dometa našega sistema. To smo ponavljali tako dolgo, da smo dobili najmanjšo oddaljenost, pri kateri je naš oddaljeni modul še lahko oddajal signal glavnemu modulu. Ta domet je znašal približno 40m, odvisen pa je bil od objektov, ki so bili med obema moduloma.

Poleg navadnega nRF24l01 modula, smo lahko testirali še modul nrf24l01, ki ima možnost priklopa zunanje antene [22]. Test smo opravili, ker je ta brezžični modul kompatibilen z našim prejšnjim modulom in smo ju lahko samo zamenjali na našem vezju. Rezultat dometa za modul z zunanjo anteno je znašal približno 90m. Na sliki 6.8 sta grafično prikazana domet obeh modulov. Z rdečim krogom je prikazan domet modula z anteno, z zeleno barvo je prikazan modul brez antene. Oranžna barva pa predstavlja naše testno okolje.

Z rezultati opravljenega testa smo bili zadovoljni, saj je bil domet prvoznega brezžičnega modula dovolj dober za pokritje našega testnega prostora. Če pa bi sistem želeli uporabljati na večji površini, bi bil verjetno bolj primeren modul z dodatno anteno.

6.3 Poraba oddaljenih modulov

Ker je glavni modul priključen direktno v električno vtičnico, na njegovo električno porabo nismo bili preveč pozorni. Pri oddaljenih moduli pa je poraba igrala kar veliko vlogo. Ti moduli so postavljeni po vrtovih, zato



Slika 6.8: Vizualizacija dometrov obeh brezžičnih modulov

je njihova čim daljša avtonomnost precej pomembna. Za izračun porabe smo vzeli 3000 mAh baterijo in izračunali, kako dolgo bi dovajala električno energijo našemu vezju. Za izračun je pomembna poraba različnih elementov vezja:

Element:	Poraba:
Arduino nano	15 mA
DHT - 11	2.5 mA
nRF24l01	15 mA
fotoupor	0.5 mA

Naš sistem ima v procesu, ko zbira in pošilja podatke, povprečno porabo 33 mA, kar ne deluje veliko, vendar pomeni, da ima naš oddaljeni modul po enačbi [19]:

$$\text{Doba baterije} = \text{kapaciteta baterije(mAh)} / \text{poraba sistema(mA)} \quad (6.1)$$

življensko dobo 90 ur, kar pomeni slabe štiri dni avtonomnosti ob porabi 33 mA. Ker pa naš modul ne zajema podatkov ves čas, ampak le enkrat na sekundo, je povprečna poraba našega oddaljenega modula 17mA. To smo izračunali tako, da približno 0.1 sekundo modul zajema podatke (poraba 33

mA), ostale 0.9 sekunde pa je v mirovanju (poraba 15 mA). Izračun po tej porabi pomeni avtonomnost približno 176 ur (dobrih sedem dni). Kar pa je vseeno premalo, da bi bil naš sistem uporabniku priročen za uporabo, še posebej, če bi imel v sistem priključenih več oddaljenih modulov.

Zato smo testirali še Arduino knjižnico JeeLib [20], za znižanje porabe naših glavnih modulov. Knjižnica je delovala brez težav. Namesto ukaza `delay(1000)`, smo v kodi, potem ko smo uvozili knjižnico, uporabili ukaz `Sleepy::loseSomeTime(1000)`. Žal pa nismo mogli izmeriti porabe sistema v mirovanju, zato smo podatek porabe vzeli iz spleta [21]. Po uporabi naj bi se poraba med čakanjem programa zmanjšala na le 0.01 mA. Knjižnica pa zmanjša tudi porabo Arduin-a nano iz 15 mA na 6 mA. Iz tega sledi, da bi imel modul porabo približno 2.7 mA. To pa bi povečalo avtonomnost sistema na 1111 ur oz. približno 46 dni, kar je že avtonomnost, ki bi bila uporabna za naš sistem.

Še ena od idej, ki bi jo lahko implementirali v našem sistemu bi bila dodaten urni modul. Ta modul bi naš oddaljeni modul sprožal vsako sekundo, sistem bi poslal podatke glavnemu modulu. Nato bi sistem spet šel v način spanja, dokler ga urni modul spet ne zbudi. Prednost take implementacije bi bila še manjša poraba, saj modul v stanju spanja ne bi porabil elektrike, urni modul pa na že zelo majhni bateriji zdrži več let.

Poglavje 7

Sklepne ugotovitve

V našem projektu smo si zadali nalogo izdelati sistem, ki bi omogočal digitalizacijo in avtomatizacijo vrtnih površin. Sistem deluje na principu glavnega modula, ki sprejema podatke oddaljenih modulov. Oddaljeni moduli so postavljeni po vrtovih, iz katerih želimo pridobivati podatke, pomembne za zdravo rast in razvoj rastlin. Sistem omogoča tudi avtomatično zalivanje vrtov, če nanj priključimo vodno črpalko in vir vode. Zalivanje se prilagaja trenutnim razmeram na vrtu in dejanski potrebi rastlin po vodi, s principom logike čakanja pa se voda enakomerno razporedi po zemlji pred ponovnim namakanjem. Sistem uporabniku nudi tudi spletno stran, preko katere lahko spremlja svoje vrtove, prav tako omogoča obveščanje preko spletne pošte v primeru, da se zemlja na vrtu preveč izsuši.

Med večjimi izboljšavami, ki bi jih lahko dodali našemu projektu, je tiskano vezje in manjši senzorji na vezju. S tem bi lahko naredili precej manjše oddaljene module. Prav tako bi spletna stran lahko imela bazo uporabnikov, tako bi vsak imel svojo spletno stran svojega vrta. Tudi avtonomnost oddaljenih modulov bi potrebovala izboljšavo, saj bi si želeli, da bi ta trajala precej dlje kot to sedaj omogoča. Ena od možnih rešitev bi bila integracija sončnih celic, ki bi polnile naše module kar med delovanjem, če bi imele zadostno količino sončne svetlobe. Lahko pa bi tudi podaljšali interval pošiljanja podatkov. Brežžična komunikacija je kar velik porabnik.

V splošnem pa smo z delovanjem našega projekta zelo zadovoljni. Sistem deluje podobno, kot smo si ga že na začetku zamislili. Najbolj nas je prepričalo testiranje našega sistema, ki smo ga implementirali v razvoj sadik čilija. Rezultati so zanesljivo govorili v prid našega projekta, saj so tako občutljive rastline, zalivane z našim sistemom zalivanja, zrasle najvišje in bile med najbolj zdravimi sadikami.

Med projektom pa smo pridobili tudi veliko znanja o naravi, še zlasti rastlinah in njihovih potrebah ter pomenu za človeka in človeštvo. Prav tako smo se naučili veliko novega in razširili strokovno znanje - kako izdelati bazo podatkov in spletno stran, brezžično komunikacijo med moduli in pošiljanje podatkov preko spleta na bazo podatkov.

Literatura

- [1] Fujitsu and PLANT DATA JAPAN to collaborate on Plant Information for Agriculture. [Online]. Dosegljivo:
<http://www.fujitsu.com/global/about/resources/news/press-releases/2015/0508-01.html>. [Dostopano 4. 12. 2015].
- [2] pH senzor kit. [Online]. Dosegljivo:
<https://www.sparkfun.com/products/10972>. [Dostopano 7. 12. 2015].
- [3] fotosinteza. [Online]. Dosegljivo:
<http://biology.clc.uc.edu/courses/bio104/photosyn.htm>. [Dostopano 7. 12. 2015].
- [4] Plant Growth Factors/faktorji za uspešno rast rastlin. [Online]. Dosegljivo:
<http://www.ext.colostate.edu/mg/gardennotes/141.html>. [Dostopano 25. 1. 2016].
- [5] K. Benyovsky Šoštarič. *Zeleni kvadrat, zdravje iz organskega vrta*. 2012, str. 45-47.
- [6] Senzor DHT-11. [Online]. Dosegljivo:
<http://www.adafruit.com/product/386> [Dostopano 7. 12. 2015].
- [7] Senzor FC-28. [Online]. Dosegljivo:
<https://www.fasttech.com/product/1380900-fc-28-soil-humidity-detection-sensor-module> [Dostopano 7. 12. 2015].

-
- [8] Foto upor LDR05. [Online]. Dosegljivo:
<http://www.resistorguide.com/photoresistor/> [Dostopano 8. 12. 2015].
- [9] Senzor zračnega pritiska BMP085. [Online]. Dosegljivo:
<https://www.sparkfun.com/products/11824> [Dostopano 8. 12. 2015].
- [10] Brežični modul nRF24L01. [Online]. Dosegljivo:
<https://http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01> [Dostopano 8. 12. 2015].
- [11] Arduino nano. [Online]. Dosegljivo:
[https:// https://www.arduino.cc/en/Main/ArduinoBoardNano](https://www.arduino.cc/en/Main/ArduinoBoardNano) [Dostopano 8. 12. 2015].
- [12] Informacije o Raspberry Pi. [Online]. Dosegljivo:
[https://https://www.raspberrypi.org/products/model-b-plus/](https://www.raspberrypi.org/products/model-b-plus/) [Dostopano 8. 12. 2015].
- [13] Dokumentacija modula nRF24L01. [Online]. Dosegljivo:
<http://maniacbug.github.io/RF24/classRF24> [Dostopano 8. 12. 2015].
- [14] ASP.NET definicija in vodič za izdelavo strežnika. [Online]. Dosegljivo:
<http://www.w3schools.com/aspnet/aspnet.asp> [Dostopano 8. 12. 2015].
- [15] Spletna stran projekta. [Online]. Dosegljivo:
<http://gardencloud.azurewebsites.net/> [Dostopano 8. 12. 2015].
- [16] Azure. [Online]. Dosegljivo:
<https://azure.microsoft.com/en-us/> [Dostopano 8. 12. 2015].
- [17] AngularJS [Online]. Dosegljivo:
<https://docs.angularjs.org/guide/introduction> [Dostopano 28. 12. 2015].
- [18] Carolina Reaper [Online]. Dosegljivo:
<http://www.guinnessworldrecords.com/world-records/hottest-chili> [Dostopano 28. 12. 2015].

-
- [19] How is Battery Life Calculated/Izračun življenske dobe baterije [Online]. Dosegljivo:
<http://jenesisproducts.com/how-is-battery-life-calculated/> [Dostopano 28. 12. 2015].
- [20] Knjižnica jeelib. [Online]. Dosegljivo:
<https://github.com/jcw/jeelib> [Dostopano 7. 1. 2016].
- [21] Zmanjšanje porabe Arduina nano. [Online]. Dosegljivo:
<https://hwstartup.wordpress.com/2013/03/11/how-to-run-an-arduino-on-a-9v-battery-for-weeks-or-months/> [Dostopano 7. 1. 2016].
- [22] Brežični modul nRF24L01 z zunanjo anteno. [Online]. Dosegljivo:
<http://www.electronicsforu.com/estore/nrf24l01p-pa-lna-v3-1-rfm05.html>
[Dostopano 7. 1. 2016].